LAUR 99-4715

Generative Analysis: Automated Concept Exploration using Multi-agent Simulations

Joe V. Holland, Randy E. Michelsen, Dennis R. Powell,
Stephen C. Upton[♦]*, and David R. Thompson*

Los Alamos National Laboratory
MS F602
P.O. Box 1663
Los Alamos, NM 87545

Keywords: agents, multi-agents, agent systems, composition, simulation

Abstract: As part of a Marine Corps sponsored project, we are developing a new multi-agent simulation technique for automatically generating concepts we call generative analysis. Generative analysis seeks to develop a new class of simulation environments that take advantage of the maturing area of complex systems science and intelligent agent research. Instead of comparing warfare concept alternatives developed by the analyst and imposed as input to drive the simulation, generative analysis techniques produce alternatives that the analyst might never have conceived. In essence, the approach is to construct an ecology of simple interacting agents (an agent is a simulation entity) and the environment within which they exist. We rely on using the power of the computer to create these ecologies of agents and imbue them with the ability to self organize, learn, communicate, and evolve over many millions of trials in order to produce the ecology that is best capable of achieving the goals and objectives given to it by the analyst. We are first exploring this new methodology in the context of tactical concepts in urban settings.

1. Introduction

Current comparative analysis and simulation methods focus on evaluating a (usually small) set of alternatives according to a set of measures of merit. For military systems analysis this involves comparing one force structure, concept or weapon system with another. For force structure analyses, forces are constrained by budget, threat, and other high level imperatives. For real-time operational planning, the constraints are time, space and the comparative state of opposing forces. In essence, analysts develop a force structure complete with equipment, doctrine, tactics, and the plan (scenario) of employment against a threat force configured in a similar way. Terrain and weather are prescribed and the scenario causes the forces to interact to produce a result. The result is iterated until convergence is achieved between the analyst's vision for the outcome of the scenario, and the computer's "bean counting" calculus that tells the analysts whether or not what they are trying to achieve is feasible. The first time through this process is called a baseline. A delta field is introduced, such as changing its equipment and tactics, and the process is repeated in order to compare it to the baseline as many times as there are alternatives. The goal is to allow decision makers to compare and contrast competing force, budget, or concept alternatives in an effort to build consensus for a decision.

One major limitation of comparative analyses and the computer simulations that support them is that they are driven by the analyst's vision or hypothesis of what the future battlefield will or should look like. This factor, coupled with very slow simulation performance, restrict the analyst from examining the many different ways in which the future battlefield *might* evolve. In addition, these systems do not provide intuition about the critical factors that make the force work effectively, other than attributing them to the

[♦] Corresponding author: Voice: (505) 667-9435; Fax: (505) 665-2017; email: upton@lanl.gov

representational calculus in attrition algorithms, etc. Many senior analysts comment that they have never been surprised by the results coming from large-scale warfighting simulations.

For the problem of future concept development, an additional major barrier is the lack of computational support for *quickly* generating and *exploring* innovative ideas. Detailed computer simulations of warfare are available, however, they are very resource consuming. The analysis process is generally resource intensive in terms of manpower, time, budget, and computational resources. Many oft used simulations require extensive attention to deriving scenario input data. Others require human in the loop resources to input decisions to the modeled forces. Other techniques, such as seminar wargames and real-world experiments also require extensive resources. These techniques are needed - however, a need also exists to enable warfighters, not just analysts, to quickly and freely explore future concepts without being burdened with a large overhead.

This capability would flexibly allow a user to generate and explore new concepts, such as: systems concepts, e.g., robotic sensors; tactical concepts, e.g., urban swarm; organizational concepts, e.g., small teams; and operational concepts, e.g., the U. S. Marine Corps' Operational Maneuver From the Sea. A user could explore the impact of various technologies on current concepts, and assess potential new concepts given the technologies. This capability could be viewed as balancing the problem of concept-pull and technology-push. It could also be viewed as bridging the gap between intuition and more extensive methods, such as detailed simulations and warfighting experiments.

This project is one thrust area of Project ALBERT, a U. S. Marine Corps Combat Development Command (MMCDC) sponsored program which "will investigate the feasibility and merit of harnessing the innovation associated with agent-based simulation and the power associated with high performance computing in order to address the “intangibles” and non-linearities associated with combat. As a baseline, ALBERT will determine if agent-based simulation can fill the current warfighting analysis gap by integrating across the human reasoning – agent based – semi-autonomous force – exercise – warfare analysis spectrum." In addition, "Project ALBERT is, inter alia, designed to exploit the present-day simulation void between human-based intuitive processes and currently established simulation methodologies, at present represented by Semi-Autonomous Forces (SAF)." (Brandstein and Horne 1998)

In this paper, the authors briefly describe the generative analysis concept and illustrate the concept by discussing a specific use case we call JIVES Town.

2. The Generative Analysis Concept

The Generative Analysis (GAn) concept originated at Los Alamos National Laboratory to address stated and perceived difficulties in applying traditional simulation techniques to pressing issues confronting military analysts in their support of decision makers. The genesis of most of these issues can be attributed to the changing emphasis, nature and uncertainty of threats since the fall of the Berlin Wall. Examples include: emphasis on different environments, e.g., conflicts in urban settings; greater uncertainty in agents who present a threat, e.g., transnational agents; and, changing nature of threat through differing means, e.g., cyberwarfare and weapons of mass destruction. Each of these confound the problem for the analyst, who now must not only contend with traditional issues, but has the additional responsibility for examining and understanding these new challenges.

The key characteristic underlying these new challenges is wide variety -- a vastly increased scope of possible threat scenarios and behaviors. And the key difficulty with applying traditional analysis techniques is manipulating these techniques to account for the wide variety. Underlying the majority of these techniques, and in particular simulations, is the assumption of limited change. This assumption was founded on the observation that, during the Cold War, the world, and the nation, was faced with a limited set of potential conflicts. Though this set was potentially very large, the boundaries were assumed fairly well defined, and the simulations and models used to assess these potential conflicts were designed accordingly. However, the effort required to modify these designs to account for the wide variety of the new challenges, is extensive. Under these considerations, therefore, we are developing and designing this new concept we call Generative Analysis with flexibility foremost in mind.

The central idea of GAn is to employ automatic generation of combinations of simulation components into scenarios to better explore the relationships among key variables in the simulation of a wide variety of concepts. (A component may represent a simulation artifact, an agent, or any entity that can generate events affecting itself, other simulated entities, or the state of the system. A simulation becomes a top-level aggregate comprised of a collection of components that interact with each other in the context of a simulated environment.) Previously this was thought expensive due either to the difficulties in modifying the simulation model or expense in running the computer to generate output. The former is mitigated by using software composition - constructing a simulation by combining collections of components (Szyperski 1998; Holland, Michelsen et al. 1999) - as the basis for model construction. The latter is still an issue more from the view of using personnel rather than computer resources. Thus there become two distinctions in assessing the outputs of a model: first is how parametric variation of a given component affect outputs. The second is how changes in component structure (not merely a variation of a component parameter) affect outputs. To clarify this distinction, consider the assessment of the effect of increasing the effective range of a main tank gun (a simple example of a systems concept). In a fixed model, we approximate the effect of increasing effective range by parameterizing tank range (often input as data values, range versus probability of kill). An alternative is to change the model of the tank system to account for changes in the overall operational characteristics, i.e., mobility, munition load, firing rate, fuel usage, etc., due to using a main gun with increased effective range. Both approaches have value, but the latter has often been assumed to be too expensive to implement using the large and complex military models currently in use. GAn does not enable infinite variation of component structure, only that which has been coded and presumably validated in the component variations for an agent. However, GAn does allow a systematic exploration of the combinatorial space of possible components, based on requirements from analysts.

Next, we illustrate the generative analysis concept through an example use case we call JIVES Town. JIVES Town is also a specific implementation of the simulation framework used by the generative analysis process.

3. JIVES Town - Example Generative Analysis Use Case

JIVES Town is an application of IVES (Integrated Virtual Environment for Simulations) (Holland, Michelsen et al. 1999) in a simplified urban military simulation domain ("J" stands for "joint" as in joint military operations). JIVES Town uses the discrete event simulation defined in the IVES framework. The protocols for the composition and simulation packages are observed and domain specific information is placed in a separate software package. Currently, JIVES Town simulates the activities of individual military infantry in a two dimensional environment which represents buildings, roads, and other infantry. The JIVES Town simulation architecture consists of two main components: the agents, which are themselves composed of other components described further below, and a simulated environment.

3.1. JIVES Town Agents

The military entities of interest in JIVES Town are modeled as agents (see, e.g., Jennings and Wooldridge 1998) and are termed JIVESAgents. The JIVESAgents structure is shown in Figure 1. The agent is composed of five systems: cognitive, weapon, propulsion, sensor, and communications (commo) based on the requirements to represent movement, attrition, communication, perception, and rational behavior. There is no inherent limit of the agent to five component systems. Other potential systems have been proposed, such as logistics, signature management, and protection management (the duals to sensor and weapon systems, respectively).

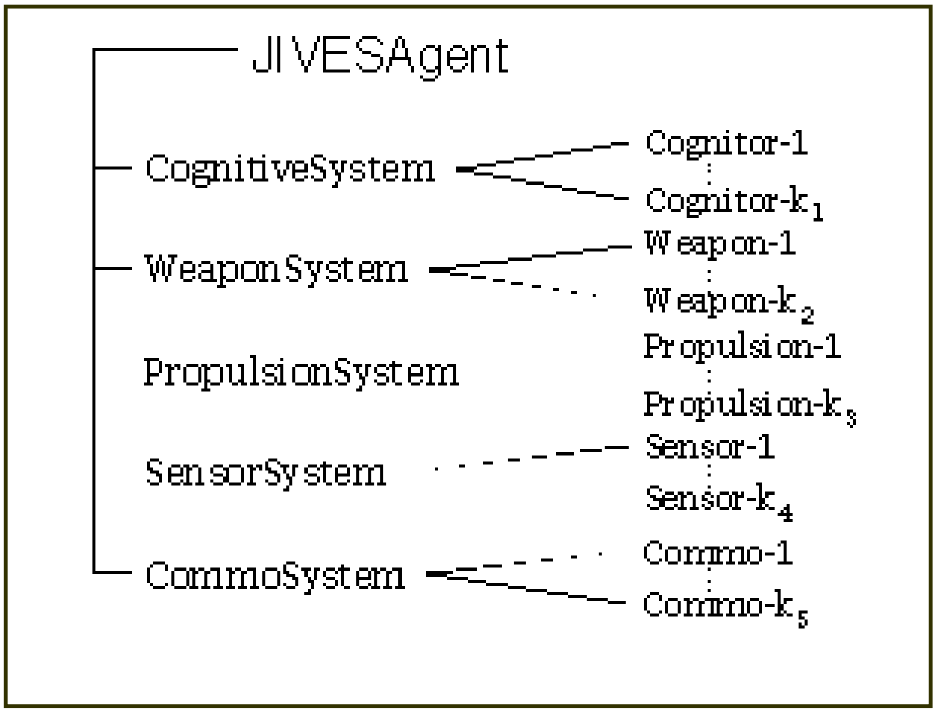

Figure 1 JivesAgent structure

These systems are containers for appropriate components, and have the responsibility for managing the components over which it has purview. The cognitive system is a container for components that support decision-making. For weapon systems, the proper components are Weapons and Munitions; for sensor systems, the components are Sensors, and so on. As examples of management functions, the weapon system can determine the best weapon and munition combination for a specified target. The propulsion system can recommend the most appropriate propulsion mode given the current conditions. Similarly, the sensor system can recommend the most appropriate sensor suite for specified classes of targets and environmental conditions. In each case, the system examines its components and recommends the components, if any, which maximize the evaluation criteria. Other management functions, like protection management, can readily be imposed if needed. The sensor system in JIVES Town is expected to monitor threat contacts, create tracks (data structures for modeling the motion of the contact), and classify tracks into threat categories.

Although the compositional nature of the agents allows for many cognitive structures, the primary one under development for the initial phase of JIVES Town is Muller's InterRap (integrated reaction and planning) architecture (Muller 1996), a version we call MilRap. This architecture specifies three layers of increasingly complex cognitive capabilities. The first layer is the Behavior Based Layer in which agents recognize situations and respond in a programmed way using patterns of behavior. This is the current level of capability for a JIVESAgent. Another way to invoke programmed behavior is to send the agent a message "commanding" the agent to invoke a specific behavior. Most often primitive behaviors are desired so that a commander can dictate arbitrary sequences of behavior and thus generate complex scripts describing desired actions. This capability is termed command-driven behavior. The utility is to allow a user or a command agent in the simulation to give orders to "subordinate" agents who will then execute those commands. Other layers of the architecture include the Local Planning Layer, and the Cooperative Planning Layer. Implementation of these layers in planned for future versions of JIVES Town.

In the current version of JIVES Town, agents maintain a set of PatternsOfBehavior (PoB) as components within the MilRap component. A PoB represents a routine task the agent can perform, based on the agent's primitive actions. A PoB includes an activation condition, a method to assess when the preconditions for the PoB are satisfied, and methods to assess whether the behavior, once invoked, was a success, failure, or encountered an exception condition requiring special processing. MoveTo is an example of a PoB in a JIVESAgent, and is based on the primitive activity of "moving one step." To move from one location to another, the agent executes a sequence of "moving one step" until the destination is encountered or there is a failure in the movement capability as represented in the propulsion system.

Constructing the agent's cognitive system with primitive behavior components will facilitate the generation of differing agent behaviors by either including or excluding various primitive components, changing the order of the components, or by changing the parameters of the components. This compositional process is the same process we use for the agent's other components.

3.2. Simulated Environment

In JIVES Town, a simulated environment is needed to support the interaction of agents. The environment component represents surface and cultural features, e.g., surface type, vegetation, roads, buildings, bridges, etc., as well as terrain elevations. Functionally, the environment supports line of sight calculation, occupancy determination, access to feature attributes, and other utility functions. Agent components such as sensors and propulsion frequently interact with the environment to update the perceptions of the agent and to constrain movement. Currently, the JIVES Town environment supports a network of roads that the agents move on, and a collection of buildings that block line of sight and movement. Planned future versions of the environment include support for 3 dimensions, to allow for movement under ground, e.g., through subways or sewer systems, and composable buildings, so that rooms and other structures can be easily incorporated.

3.3. Example Concept Exploration

We define concept to mean any notion or idea in, but not limited to, tactics, organizational structure, organizational processes, or material system. By their very definition, concepts are typically ill-defined, imprecise and vague, demanding little detail. We define concept exploration to be the unfettered and unconstrained - out-of-the-box if you will - look over a broad range of possibilities for these concepts over a broad range of conditions.

In our current vision of Generative Analysis, we would trade detail in specification of the scenario and its components (the agents, their components, and the environment components), and the consequent very limited number of runs, with a very large number of runs, potentially in the millions, over the scenario space with limited, less detailed models of the environment and scenario components. This larger sampling would hopefully gives us greater insight into the factors which are most important to concept effectiveness. If we then wanted to conduct a more detailed analysis, we might have better insight into which areas of the scenario space to focus our attention.

Next, we give a more detailed example of what we mean by looking at a typical system concept exploration. Less detailed examples of other concept explorations follow.

A typical system concept exploration might address questions similar to:

1. Given a fixed force structure and tactics, how does replacing a system with different alternatives affect urban operations? This is the essence of an Analysis of Alternatives. Some examples might be looking at the effect of differing AAAV's, sensors, information systems, tank guns, etc..
2. Given a fixed force structure and tactics, how does adding a new system affect urban operations? For example, we may want to add a new sensor to the squad's Table of Equipment and examine the impact on mission effectiveness given the new equipment. This example is explored in more detail below.
3. Given a fixed force structure and tactics, how do differing weapon mixes affect urban operations? One option might have a different mix of assault weapons, like anti-tank, mortars, machine guns, etc., but all the options are built out of a fixed set of weapons.

The goal of this type of analysis would be to examine the impact of a new system concept. This new system might be a proposed replacement for a current system, e.g., replacing the M-16 with the Super-Duper Urban Combat Destroyer, or adding this new system to a unit's Table of Equipment (T/E), e.g., Micro UAVs for reconnaissance. The former would be indicative of an Analysis of Alternatives.

Suppose we take the example of adding a new system, and for the sake of discussion, suppose we want to examine the impact on mission effectiveness of adding Micro UAVs to a squad's T/E. Performing an analysis by replacing a system would be similar.

Micro UAVs are small, insect-like UAVs which have very limited range and performance. Our goal in this analysis is to examine how these Micro UAVs add to the mission effectiveness of a squad operating in the urban environment. After development of Essential Elements of Analysis (EEA) and Measures of Effectiveness (MOE), a traditional comparative analysis using a detailed simulation consists of selecting a small (2 or 3) set of "representative" scenarios and then manually develop (e.g., through a series of wargames) a set of tactics for the squads with Micro UAVs. The "representative" scenarios might contain "mixtures" of urban environments, e.g., industrial, suburban, city center, ports, etc., to determine if these factors impact squad effectiveness with and without Micro UAVs. Each of the scenarios would also probably include a fixed structure and tactics for the Red force, e.g., specific numbers, positions, and types of weapons. A considerable amount of time would also be spent on identifying and collecting data for the simulation. Detailed models for the Micro UAVs would probably need to be developed, in addition to any additional required changes to the simulation, if for example, we couldn't modify an existing object, e.g., an aircraft to "model" the effects of the Micro UAV. Analyses would then be performed on this relatively small number of simulation runs, possibly in conjunction with an abridged sensitivity analysis.

How might a Generative Analysis look? As in the analysis above, we would probably start with development of EEAs and MOEs; however, we would not feel constrained to develop them fully prior to an exploration of the concept. One of the premises underlying the use of Generative Analysis is that we don't know all of the boundaries of the analysis, i.e., we don't know what we don't know, and so we begin our exploration with incomplete, imprecise, and uncertain knowledge.

Next, we would construct an agent model of the Micro UAV, i.e., determining in a coarse fashion, its behavior, such as movement, and its attributes, e.g., how long can it fly, its rate of movement, what kinds of sensors it has, to whom and where does it sends its information, etc.. We fill out any parameters with data we know or postulate, in addition to specifying a range of values that it *could* have. Any uncertainty in the parameters will eventually be sampled over, and thus indicate whether that parameter is important to overall effectiveness. Since all JIVES Town agents have similar structure, the work of building a Micro UAV agent boils down to adding a propulsion, sensor, and communication component, and then filling out the parameters for each of those components. Additionally, we might need to change the cognitive system of the agent to properly reflect its behavior. Once that work is completed, which could take as little as a few hours, a Micro UAV agent is added to a repository, for potential reuse by other simulations and analysts.

We then need to decide on the bounds of the scenario space in which to conduct our exploration of this systems concept. This decision is based on the objectives of the analysis, and also on the amount of resources (e.g., computational, analysts) and time allotted for conducting the analysis. Is it an unconstrained or constrained exploration? If constrained, in what way? Do we examine only inner city environments? suburban? industrial zones? Are the Blue force structure and tactics fixed? Is Red similarly constrained?

Once we have determined the bounds on the scenario space, we then turn the sampling of the scenario space over to the computer. From this broad specification given to it by the analyst, the computer then generates points in this scenario space. Each point in the scenario space is a <u>particular</u> scenario, e.g., a specific number of Blue and Red forces, the tactics and operating procedures used by the Blue and Red forces, a specific number of Micro UAVs, the UAVs specific characteristics, etc.. Depending on the objectives of the analysis, there may be several runs of the simulation with this particular scenario, e.g., if learning is required on the part of the infantry agents using the Micro UAVs, or finding the "best" behavior of the Micro UAVs. The sequence or procedure in which the computer generates the points in scenario space is also a function of the analysis requirements. If the goal is an unconstrained exploration, we might use a space filling experimental design. If the goal is to optimize some metric, we might use a natural nonlinear optimization technique, such as simulated annealing or a genetic algorithm. (Back, Fogel et al. 1997; Mitchell 1997) As is probably obvious, the number of differing combinations of components, in addition to the various parametric changes, result in a very large space to search. However, given the current and projected availability of very fast computers, fast running simulations, and new and emerging techniques in the statistical and optimization sciences, we are confident this generative analysis approach will yield additional insight that can not be delivered using traditional techniques. In the same vein, we do not expect to supplant traditional techniques, but to complement them.

The computer then continues to sample over the scenario space through a compositional variation, i.e., changing the <u>numbers</u> and <u>types</u> of components in a simulation, and a parametric variation, i.e., changing the <u>values</u> of the attributes of those components. This sampling is completed upon satisfying an analyst specified termination condition. The analyst then uses the to-be-developed visualization and data analysis tools, or additional tools at hand, to address his question or develop some insights into the behavior of the system with and without the new addition.

Using the developing Data Farming techniques (Brandstein and Horne 1998), the analyst could then modify his assumptions, change the bounds on the scenario space to search, or ask other questions.

Other concept explorations are similar in methodology, structure and content as the system concept exploration detailed above. The major difference in the system concept exploration and other concept explorations, such as tactical and organizational concepts, is the degree to which we hold elements of the agents and their component's behavior and attributes fixed. For example, if we assume tactics are modeled by the embedded rules in the cognitive system of the agent and their subsequent emergence within a collective of agents (possibly an oversimplified assumption), then to examine a tactical concept, we hold fixed the system components, and vary only these rules.

Tactical concepts are interrelated to specific systems, the environment, and threat capabilities. For example, infantry tactics are very different when the environment is in open terrain (e.g., desert) or restrictive terrain (e.g., urban or jungle). If the threat has significant anti-infantry capability, then the infantry tactics possibly change. For tactical concepts, it is these relationship we wish to explore, and concurrently, have the agents in their synthetic environment explore. The goal is to see if these changes do, in fact, occur, and identify the causal factors. A typical tactical concept exploration might address questions similar to "Given a fixed force structure, how do differing tactics, such as Urban Swarm, Urban Penetration, and Urban Thrust affect operations?" (Wood 1998) In this case, we have humans who have proposed some initial tactics to investigate.

A typical organizational concept exploration might address questions such as "How do different C2 structures affect urban operations?" In this case, again, we might have human experts propose several different structures in which to evaluate. The ideal case, again, is to have the computer generate and explore different options. An example of this type of exploration might be a change in squad organization, either by modifying the number of Marines in a squad, the composition of the fire teams or how that squad requests resources, such as fire support. Other, more complicated organizational structures can be attempted as we learn how to move up the scale in the number, types and organizational complexity of the agents.

4. Conclusion

Hopefully, we have presented a sufficient case to consider Generative Analysis as a new paradigm, and tool, for generating and exploring new concepts. However, the Generative Analysis concept is still under development and the only thing we are sure of is that we have only scratched the surface of the capabilities, the requirements, and the potential challenges.

One such challenge we recognize, and is probably obvious from this paper, is the reliance on very high performance computers with which to generate, and analyze, the simulations. With all the possible combinations in scenario space, even with a limited sampling and exploration, the number of simulations generated could number in the millions or more.

5. References

Back, T., D. B. Fogel, et al., Eds. (1997). **Handbook of Evolutionary Computation**. Philadelphia, PA, Institute of Physics Publishing.

Brandstein, A. G. and G. E. Horne (1998). "Data Farming: A Meta-technique for Research in the 21st Century". Headquarters, U. S. Marine Corps. Maneuver Warfare Science 1998. Quantico, VA.

Brandstein, A. G. and G. E. Horne (1998). "Project ALBERT Terms of Reference". Marine Corps Combat Development Command. Quantico, VA.

Holland, J. V., R. E. Michelsen, et al. (1999). **Composing Simulations Using Persistent Software Components**. 1999 Advanced Simulation Technologies Conference (ASTC '99), San Diego, CA, The Society for Computer Simulation International.

Jennings, N. R. and M. J. Wooldridge, Eds. (1998). **Agent Technology: Foundations, Applications, and Markets**. New York, NY, Springer-Verlag.

Mitchell, M. (1997). **An Introduction to Genetic Algorithms**. Cambridge, MA, The MIT Press.

Muller, J. P. (1996). **The Design of Intelligent Agents: A Layered Approach**. Berlin, Springer-Verlag.

Szyperski, C. (1998). **Component Software: Beyond Object-Oriented Programming**. Reading, MA, Addison Wesley Longman.

Wood, A. W. (1998). "Urban Warrior Conceptual Experimental Framework, Version 1-5". Marine Corps Combat Development Command. Quantico, VA.